

Sensor Basics Utility Library

The Sensor Basics utility library provides small validation and conversion helpers shared by the sensor drivers and the main application. These are basic features that could be used if required, but were made in "spare time".

Source Code Location

Files:

- `components/sensor_board/sensor_basics/sensor_basics.h` (function declarations and documentation)
- `components/sensor_board/sensor_basics/sensor_basics.c` (implementation)

Dependencies:

- `result.h` (standard result/error code definitions and the TRY macro)
- `stdint.h` (integer type definitions)

pH Validation

`validate_ph_value()`

Validates that a pH value is within the acceptable range (0 to 14).

```
result_t validate_ph_value(float ph_value) {
    if (ph_value >= 0.0f && ph_value <= 14.0f) {
        return RESULT_OK;
    }
    return RESULT_ERR_INVALID_DATA;
}
```

Return values: `RESULT_OK` when the value is within range, `RESULT_ERR_INVALID_DATA` otherwise.

Note: `ph_sensor_update()` already clamps its output to 0..14, so this check only fails for values that bypass the driver (for example raw values received over the network).

IMU (Accelerometer) Validation

validate_accelerometer_value()

Validates one accelerometer axis. Valid range is -160.0 to +160.0 m/s², which corresponds to a typical ± 16 g sensor range.

```
result_t validate_accelerometer_value(float accel_value) {
    if (accel_value >= -160.0f && accel_value <= 160.0f) {
        return RESULT_OK;
    }
    return RESULT_ERR_INVALID_DATA;
}
```

validate_imu_data()

Validates all three accelerometer axes at once using the TRY macro for early return on the first invalid axis.

```
result_t validate_imu_data(float accel_x, float accel_y, float accel_z) {
    TRY(validate_accelerometer_value(accel_x));
    TRY(validate_accelerometer_value(accel_y));
    TRY(validate_accelerometer_value(accel_z));
    return RESULT_OK;
}
```

Note: the IMU driver has its own richer validators (imu_validate_accelerometer_range, imu_validate_gyroscope_range, imu_validate_magnetometer_range) which the main loop uses. See the imu component.

Conversion Functions (declared, currently inactive)

The header declares four unit conversion helpers. Their implementations exist in sensor_basics.c but are commented out, so linking against them fails until they are re-enabled.

```

/* Temperature: F = C * 9/5 + 32, C = (F - 32) * 5/9 */
result_t celsius_to_fahrenheit(float celsius, float *fahrenheit);
result_t fahrenheit_to_celsius(float fahrenheit, float *celsius);

/* Pressure: psi = bar * 14.5038, bar = psi / 14.5038 */
result_t bar_to_psi(float bar, float *psi);
result_t psi_to_bar(float psi, float *bar);

```

Each returns RESULT_OK on success or RESULT_ERR_INVALID_ARG when the output pointer is NULL.

Implementation Status

Currently implemented (active):

- validate_ph_value()
- validate_accelerometer_value()
- validate_imu_data()

Currently commented out (inactive):

- celsius_to_fahrenheit()
- fahrenheit_to_celsius()
- bar_to_psi()
- psi_to_bar()

Note: earlier drafts of this page documented GPS validation helpers (latitude, longitude, HDOP, satellite count). These functions do not exist in the current library since there is no GPS driver on the sensor board.

Error Handling Pattern

All validation functions follow the same pattern used across the firmware:

```

if (validate_ph_value(ph_value) == RESULT_OK) {
    diagnostics.ph_sensor.state = SensorState_SENSOR_OPERATING;
    diagnostics.ph_sensor.error_code = PHErrorCode_PH_NO_ERROR;
} else {
    diagnostics.ph_sensor.state = SensorState_SENSOR_ERROR;
    diagnostics.ph_sensor.error_code = PHErrorCode_PH_INVALID_DATA;
}

```

```
}
```

Testing

Test suite location: `test/sensor_board/test_sensor_basics/`

```
# Run only the sensor_basics tests
pio test -e sensor_board -f test_sensor_basics

# Run with verbose output
pio test -e sensor_board -f test_sensor_basics -v
```

Current coverage: accelerometer boundary values (± 160.0 accepted, ± 160.1 rejected) and multi-axis combination. The temperature and pressure conversion tests exist in the file but are commented out together with their implementations.

Revision #3

Created 2026-04-14 17:10:12 UTC by Shishir Nambiar

Updated 2026-07-08 14:25:58 UTC by Shishir Nambiar